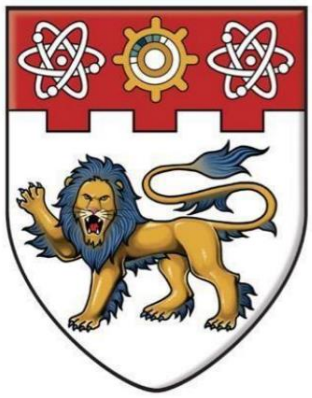
Machine learning based face expression recognition

Paing Thu Thu Aung

2022



NANYANG TECHNOLOGICAL UNIVERSITY

School of Electrical and Electronic Engineering

Academic Year 2020/21

Semester 2

EEE Final Year Project

Machine Learning Based Face Expression Recognition

(P3042-202)

Submitted by: Paing Thu Thu Aung (U1820473L)

Main Supervisor: Assoc Prof Jiang Xudong

Contents

List of Figures

List of Tables

Abstract

Face expression recognition is an active research area in the past two decades. Many attempts have been made to understand how human beings perceive human faces. It is widely accepted that face recognition may rely on both componential cues (such as eyes, mouth, nose, and cheeks) and non-componential/holistic cues (considering the face as whole rather than as separate parts). However, how these cues should be optimally integrated remains unclear. Most state-of-the-art technologies of face expression recognition employ either componential cues or holistic information. Their recognition performance is therefore limited.

This project investigates ways to integrate componential and holistic cues. We deployed a pretrained facial landmark detector to locate 68 landmarks of a face, to extract 8 individual facial components. Next, we utilized a convolutional network (CNN) to extract and learn relevant features from the facial and 8 componential images. Moreover, we deployed a CatBoost classifier to classify the landmark coordinates. Finally, we deployed soft and hard voting to combine all the predictions of the 10 trained models together. The soft voting approach achieved an accuracy of 63.87%, which is comparable to some existing method, considering we deployed fewer data for training. The creative approach may potentially lead to a better face expression recognition technology that outperforms current existing methods.

Acknowledgement

I'd like to thank Assoc Prof. Jiang Xudong for giving me the opportunity to work on this fantastic project of machine learning based face expression recognition. Machine learning and facial expression recognition have always piqued my interest. His information and papers were extremely useful to me. In a short period of time, I learned a great deal from the papers shared by him.

Although this project is being carried out with great care and enthusiasm, there are some flaws. Feedback is welcome, and I hope this report is useful to readers who want to understand my approach to this project.

Paing Thu Thu Aung (Sally Soo)

Chapter 1 Introduction

Face expressions convey non-verbal communication cues that play an important role in social situations. During interpersonal communication, humans give and receive non-verbal cues even when silent. The human face is the most expressive of them all, capable of conveying a wide range of emotions. It is stated that a person's face reveals his current state of mind.

Humans naturally exhibit their personal emotions in social contexts. Within a split second, subtle variations in facial expression can portray the fluctuation of numerous emotions in a person's head. Mutual understanding and trust can be built through having a clear comprehension of one other's emotions. Individuals and corporations alike benefit from an understanding of expression.

Application

To conduct market research, users' reactions are traditionally observed while interacting with a brand or a product. Unfortunately, people may unconsciously transmit perplexing or unfavourable non-verbal cues. While analysing behavioural approaches is time-consuming, it is also not scalable, and accuracy varies based on the surveyor's experience and viewpoint.

In such cases, facial expression recognition technology can come in handy. It can produce raw emotional responses that reveal crucial information about a target audience's feelings about a marketing message, product, or brand. It enables businesses to conduct market research and automatically measure moment-to-moment emotional facial expressions, making it simple to combine the results and determine the efficacy of any business content.

Facial expressions, unlike some other kinds of non-verbal communication, are universal. Face expressions can be classified as angry, disgust, fear, happy, neutral, sad and surprise. Various studies have been conducted over the years. Due to the complexity and variety of human facial expression emotion components, traditional facial expression emotion identification technology suffers from insufficient feature extraction and susceptibility to external environmental impacts.

Current Technology and Methods

Facial expression recognition has been technologized by different approaches. To the best of the author’s knowledge, most literature focuses on building deeper neural networks with millions to billions of parameters such as the VGGNet, ResNet, Inception, DeepEmotion, and so on, for recent years. All these approaches deployed different variation of convolutional neural networks (CNN), with different number of layers and filter size. One consistent aspect of these approaches is that they input the image directly into the deep learning models, without additional or further pre-processing techniques. Therefore, they do not consider the features of individual component within a face structure. However, a study from Fujitsu

Laboratories claimed that introducing componential features in addition to facial features can improve facial recognition performance.

Objective

The objective of this project is to research and investigate ways to optimally integrate both holistic and componential cues of face expressions. This is due to with human inspection, we can tell a person's emotions even when partial components of a face. For example, if a person's eyebrows are shrunk and sides of the mouth are in grin position, we can tell the person is angry. Another is if we only see a person's mouth area and the corners of the mouth are turning upwards and teeth are showing, we can tell that the person is happy. This project focuses on extracting componential features and deploying machine learning techniques to classify facial emotions, to evaluate if it is possible to produce an ideal with potential of contesting with the current state-of-the art methods.

Chapter 2 Dataset

Datasets are the foundation for training, evaluating, and benchmarking machine and deep learning models, and they have played a critical role in the field. Machine and deep learning data analysis employs algorithms to continuously improve itself over time, but good data quality is required for these models to function properly. Quality datasets are essential for model sculpting or neural network training. Labelled data is important in machine learning because it simplifies the work of the machine learning program in many cases. To learn how to classify, training data must be fed into the machine learning algorithm, followed by testing data to evaluate if the model is properly fitted and correctly predicting the labels.

FER2013 Overview

As machine and deep learning algorithms are heavily reliant on data, it is important that there are sufficient and meaningful data. To have meaningful data, the quality of the dataset must be taken in account. However, data may contain flaws and noise. Of all the public datasets available across the web, we selected FER2013 dataset as the dataset in this project. The main reason is that it is one of the largest public datasets available, and it has been the subject of numerous research projects over the years. According to Kaggle, this dataset was created as part of an ongoing research project by Pierre-Luc Carrier and Aaron Courville [1].

The dataset is approximately 57MB in size and is made up of grayscale images of 48 by 48 pixel facial images. The faces have been automatically registered to be centred and to take up roughly the same amount of space in each image. The data has already been divided into test and training datasets. The information is labelled into seven categories: angry, disgust, fear, happy, sad, surprise, and neutral. The testing set is used for evaluating the model for accuracy purpose, while the training set is used to train the machine or deep learning model.

The training set contains 28,709 images, while the test set contains 3,589 images. The distribution of examples within the known classes is slightly skewed. Both the training and testing datasets have unequal class distributions. In the training dataset, for instance, happy expression has the most files (7215) and disgust expression has the fewest (436). Similarly, in the testing dataset, sad expression has the most files (1247), while disgust expression has the fewest (111). Figure 1 shows the overall view directory of the dataset.

Figure 1: Overview of the directories in the FER2013 dataset.

FER2013 Dataset Complications

The data distribution in the FER2013 dataset raises the issue of class imbalanced classification. Most classification predictive model machine learning algorithms are designed and demonstrated on problems with an equal number of classes. This means that a naive model application may focus solely on learning the characteristics of the majority class, ignoring examples from the minority class, which is, in fact, of greater interest and yields more valuable predictions. This issue can be addressed using various techniques. One of them could be data augmentation. Table 1 below summarizes the size of various datasets in each of their classes.

Table 1: Number of testing and training samples for each emotional class from the FER2013 dataset.

Class	No of Testing Files	No of Training Files
Angry	958	3995
Disgust	111	436
Fear	1024	4097
Happy	1774	7215
Neutral	1233	4965
Sad	1247	4830
Surprise	831	3171

Data Extraction

After studying the dataset, required data is extracted from the dataset. As the dataset provides images, it makes it very convenient for us to extract both componential, holistic and hybrid data. Figure 2 explains the process of overall data extraction.

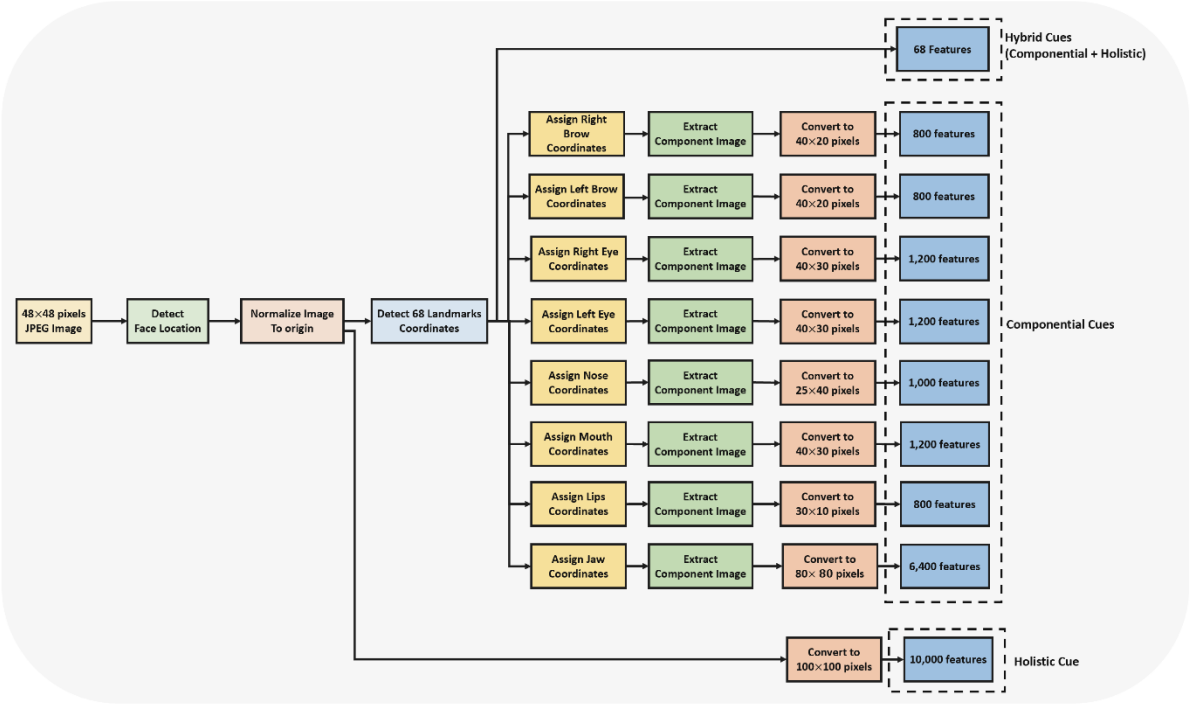


Figure 2: Proposed data extraction for the FER2013 dataset.

Features Extraction

First, face location is detected in all the images and normalised to the origin. After that, for holistic approach, the face is treated as a whole and resized into 100 by 100 pixels. Then, 68 landmarks coordinates are detected using 68-dlib landmark pretrained model. From the coordinate points, the images of each component (right brow, left brow, right eye, left eye, nose, lips, mouth, and jaw) are extracted. From the images, the features are exacted for componential wise.

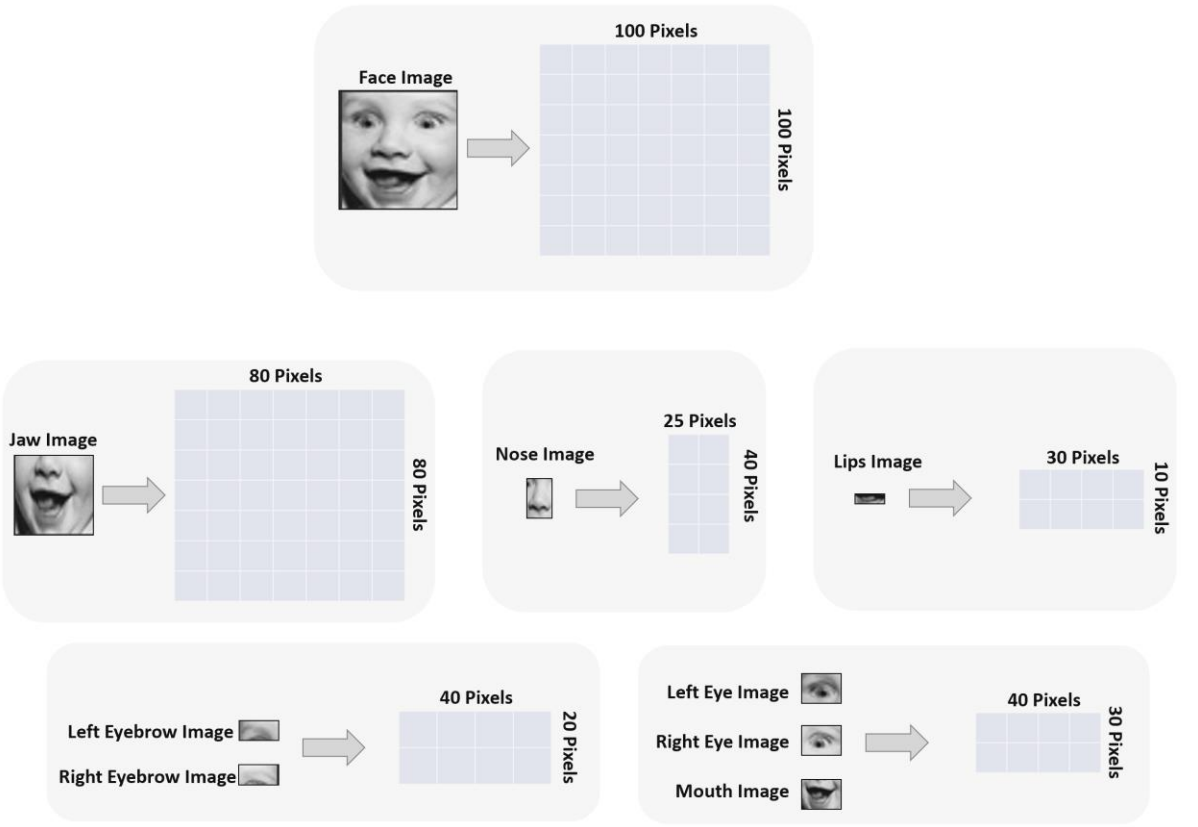


Figure 3: Proposed feature data resizing for each component by pixels for the FER2013 dataset.

Landmark Extraction

The coordinates of landmarks are also extracted as feature for hybrid use as well. All the data are separately in different csv files with proper labels, generating 10 csv files as database in this case. Refer to the Figure 4 for the landmarks detected.

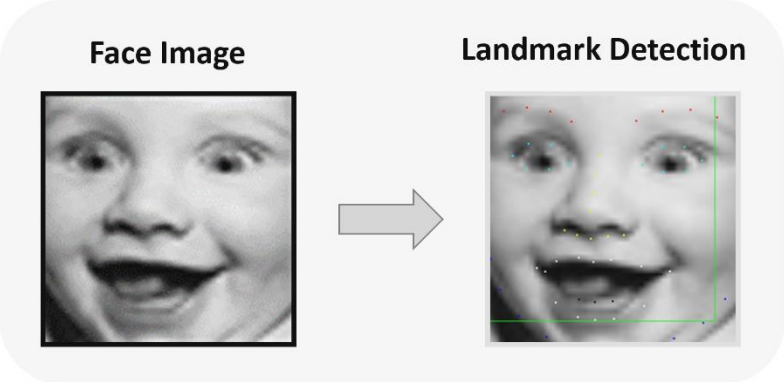


Figure 4: Proposed face landmark data extraction for FER2013 dataset.

Data Loss

During data extraction, some of the data are lost as the entirely of the face may not be within the image. For instance, the face may be partially covered with hands, or the eyes may be covered with sunglasses. Such situation may create occlusion problem, and we will not be able to extract the componential information from the image. In this case, the 68-dlib landmark detector may not be unable to detect certain landmark points on the edges of the face and drop the data. Table 2 consolidates the data loss during data extraction.

Table 2: Data loss in the testing and training data from the FER2013 dataset.

Class	No of Extracted Testing Files	Testing Data Loss (%)	No of Extracted Training Files	Training Data Loss (%)
Angry	596	38	2440	39
Disgust	87	3	321	26
Fear	581	46	2302	44
Happy	1310	48	5088	29
Neutral	853	40	3396	32
Sad	589	69	2483	49
Surprise	528	32	2123	33

Chapter 3 Methods

Python is used in this project as it contains many prebuilt libraries that can perform the necessary tasks. Sypder IDE is used for coding, debugging, and troubleshooting. The libraries that are used are listed below.

Feature Extraction

To detection and extract facial features, we implemented the following libraries.

- **OpenCV** is a computer vision application library which supports libraries such as Scikit-Learn, TensorFlow and Keras. These libraries will be used for emotion classification to identify the face expressions from the input dataset.

- **Dlib** is a library written in C++ toolkit containing machine learning algorithms and tools to solve real world problems. In this project, it will be used to do facial mapping and identify landmarks of the face. From the landmarks, face components will be identified, and data will be extracted for later use. The model to determine the landmark is a pretrained model using data from the IBUG dataset. In this work, we will be using the pretrained model directly to perform our landmark localization.

- **Dlib's 5-point facial landmark detector** [2] assigns two points to the left eye's corners, two points to the right eye's corners, and one point to the nose This detector is used to align the faces. However, after several attempts, it takes too long to align the faces, so this was not used in the project's final stage.

- **Dlib's 68-point facial landmark detector** [3] recognizes 68 points on a human face (coordinates (x,y)). The area around the eyes, brows, nose, mouth, chin, and jaw is pinpointed using these points. Figure 5 shows the landmark locations.

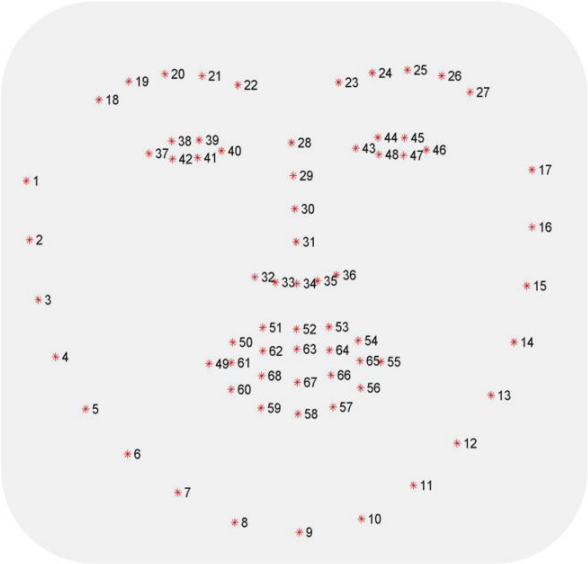


Figure 5: Dlib's 68-point facial landmark positions and indices.

Other libraries such as **NumPy and Pandas** will also be used to perform effective scientific computing in backend. NumPy will be used in manoeuvring numerical arrays and Pandas will be used in manipulating tabular data in terms of data frames and series for data preparation.

Feature Learning and Classification

- **TensorFlow** is used to quickly and easily build and train ML models using intuitive high-level APIs like **Keras** and eager execution, which allows for immediate model iteration and simple debugging.
- **CatBoost** [4] is open-source library for gradient boosting on decision trees. It is used to analyse the 68 landmark points for componential data analysis.
- **Scikit-learn** is a package that provides efficient versions of a wide range of common algorithms.
- **Matplotlib and Seaborn** are used to visualize the results data. One of the applications includes creating confusion matrix to determine how good the results are.

Proposed Algorithm Pipeline

The proposed facial detection algorithm pipeline is displayed in Figure 6. First, the webcam will take in the live feed video. Next, for every frame, we attempt to locate the presence of any face in the image. If a face is detected, we resize the face to 100 by 100 pixels, and perform 68-point landmark detection to obtain the positions of each facial components.

Then, we perform three groups of classification:

1) Facial image classification.
2) Component image classification.
3) Landmark position classification.

The facial image classification is as follows. First, we normalize the face that was originally detected into a BW image, with values between 0 and 1 for every pixel. Next, we perform classification on the image with a CNN model, to obtain probability outputs between 0 and 1 for each emotional class.

Meanwhile, with the positions of the 68 landmarks and the respective indices of the 8 facial components, we extract the image of 8 components. Next, like the facial image classification, we normalize all the image, and perform 8 separate CNN classification for each 8 facial component images to obtain 8 probability outputs between 0 and 1 for each emotional class.

Lastly, the landmark classification uses only the positions of the 68 landmarks for both x and y axis to perform emotional prediction. Firstly, we normalize the landmark values to the origin of the original face image. Next, we deployed a CatBoost classifier to classify the landmark values to obtain probability outputs between 0 and 1 for each emotional class.

Finally, with the 10 probability outputs obtained from the three classification groups, we apply a soft voting to obtain the final predictions. The final prediction is the maximum probability of a given class after the soft voting stage.

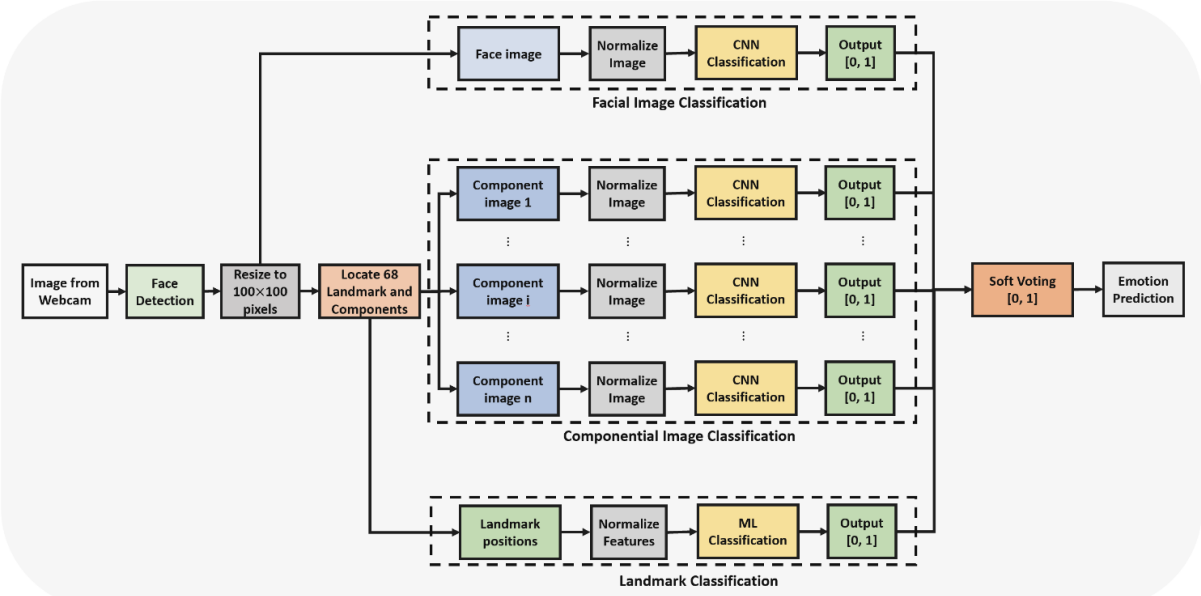


Figure 6: Proposed algorithm of the facial expression detection pipeline. The pipeline consists of several CNN models to classify different facial components and a soft voting is applicated to combine all their outputs to form a final prediction.

Machine Learning Basic Theory

Convolution

Convolution is a technique for altering an image by applying a kernel to each pixel and its local neighbours across the entire image. [5] The kernel is a value matrix whose size and values influence the transformation impact of the convolution process. The Convolution Process is comprised of the following steps.

1) The Kernel Matrix is placed over each pixel of the image, and each value of the Kernel is multiplied by the pixel it is over.
2) This procedure is repeated throughout the image.

Convolution is used for feature extraction such as edges, corners and so on. Several types of filters can be used for convolution. In the Figure 7, a 3 by 3 kernel of edge filter is convoluted over a 5 by 5 source image. The kernel's Centre Element is positioned over the source pixel. After that, the source pixel is replaced with a weighted sum of itself and the pixels around it. The output is placed in the destination pixel value.

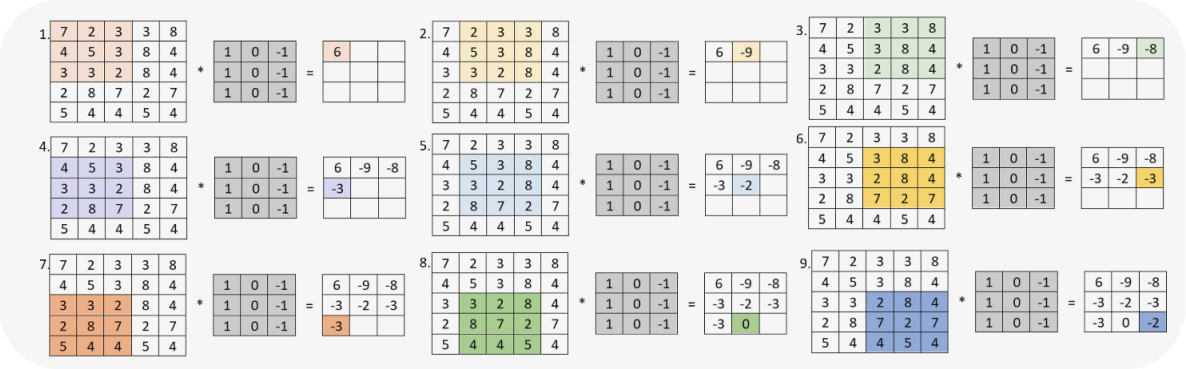


Figure 7: Convolution filter without padding.

This will be applied to all the data in our dataset. The difference is, with machine learning from Keras, kernel will be randomly assigned to fit, and input greyscale image will be convoluted.

Max-Pooling

Max-pooling calculates the maximum value in each patch of each feature map and forwards the best features, hence reducing the size of the image to reduce computation power. [6]In this example, 4 by 4 matrix is reduced to 2 by 2 matrix. This is beneficial to reduce computation intensiveness and overfitting. Overfitting occurs when good accuracy for training is achieved but bad accuracy for testing occurs. Max pooling can be increased to resolve the issue.

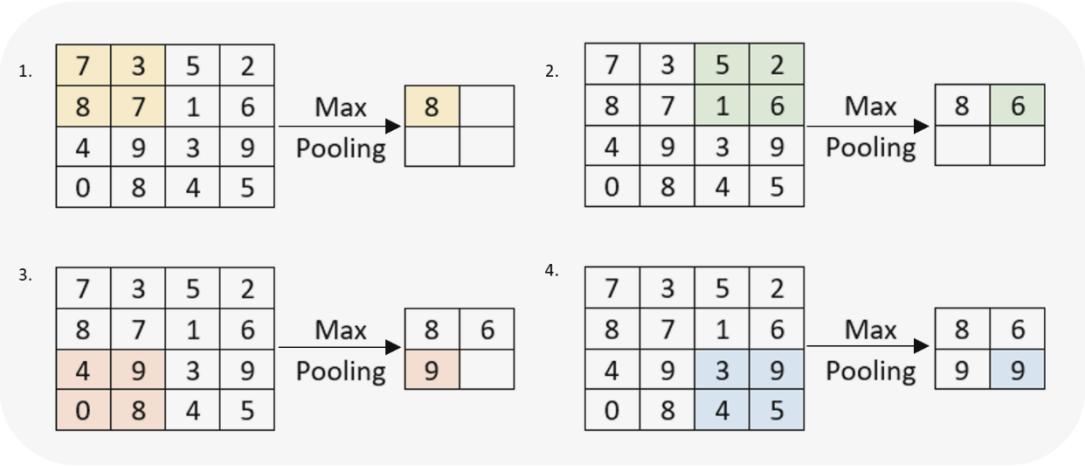


Figure 8: Max-Pooling 4 by 4 matrix.

Rectified Linear Unit Activation Layer

Activation function introduces non-linearity by converting negative value to zero. For the negative input values, ReLU makes the neuron to not get activated. Since only a certain number of neurons are activated, the ReLU makes the model computationally efficient. ReLU can be computed as:

$$ReLU(x) = \max(0, x) \quad (1)$$

Where x is the input.

Flatten, Dense and Drop Out

The process of converting data into a one-dimensional array for use in the next layer is known as flattening [7]. We flatten the output of the convolutional layers to create a single long feature vector. Later, the flattened data will be fed into the dense layer.

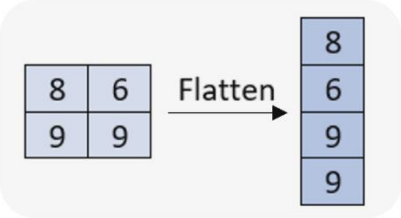


Figure 9: Flattening 2 by 2 matrix.

The dense layer is a simple layer of neurons in which each neuron receives input from all neurons in the preceding layer. Dense Layers are used to identify images based on the output of convolutional layers [7].

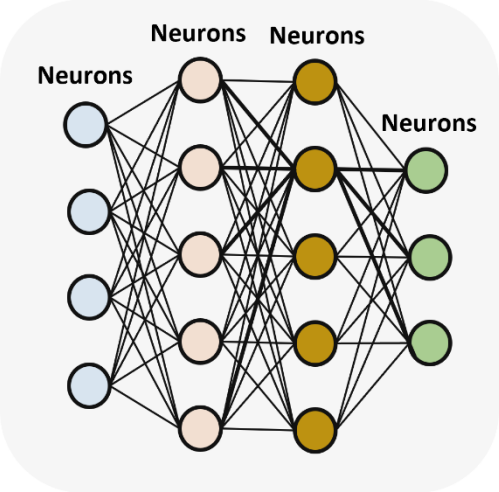


Figure 10: Dense layer.

The dropout technique can be used to avoid overfitting. Dropout operates at each update of the training phase by setting the outgoing edges of hidden units (neurons that make up hidden layers) to 0 [7]. Figure 11 below depicts some randomly dropped neurons.

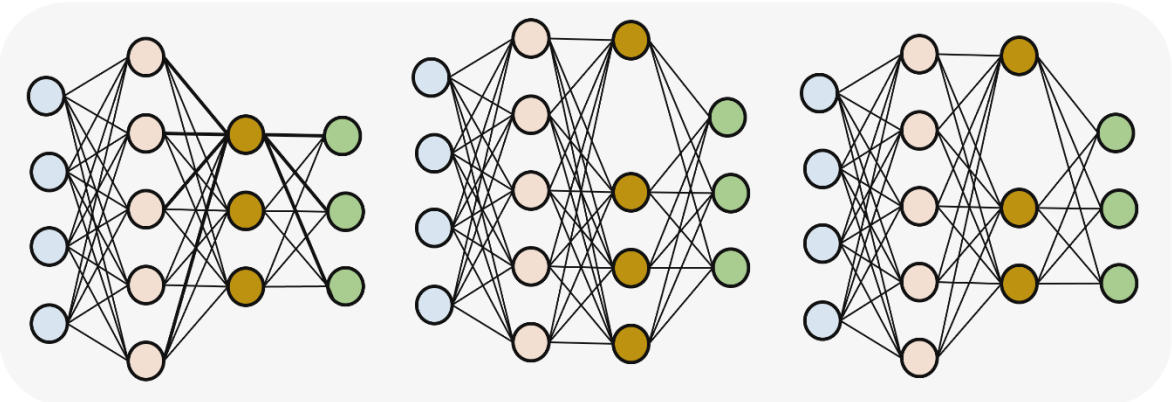


Figure 11: Drop-Out layer.

Softmax

Softmax is a function that rescales an output to a value between 0 and 1 [8]. It outputs the probability of a given class intuitively, as probabilities also range from 0 to 1. Non-normalized inputs are converted into a set of exponentiated and normalized probabilities by the softmax. When there are more than two outcome classes, the softmax is typically used to generalize logistic regression in multi-class classification problems.

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{2}$$

Where σ is the softmax, z is the input vector, e^{z_i} is the standard exponential function for input vector, K is number of classes in the multi-class classifier and e^{z_j} is the standard exponential function for output vector.

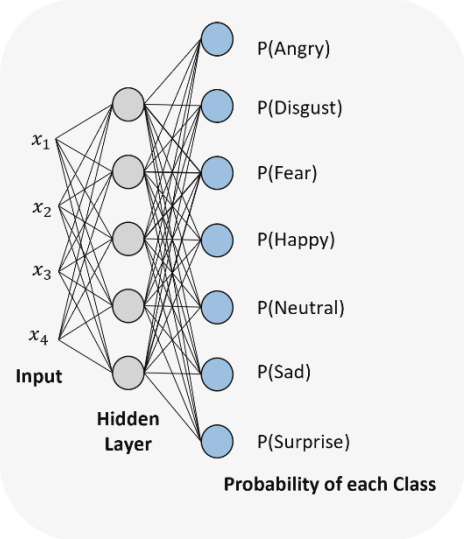


Figure 12: Softmax layer.

Voting Classifier

All the cues in the extracted data will have different models which gives 1 holistic model, 8 componential models and 1 landmark model in the end. The results from those models will be combined to do voting to give a one class. This integration of all the models will be achieved by soft and hard vote.

The predicted output class in hard voting is the class with the most votes. The output class in soft voting is the prediction based on the average of the probabilities assigned to that class [9].

Models Architecture

Convolution Neural Network (CNN)

The convolutional neural network (CNN) consists of different layers, such as the convolutional layer, max pooling, flatten, dense, dropout and softmax. The convolutional layer and the max pooling help in feature extraction while the flatten, dense, dropout, and softmax are for classification.

CNNs are typically deployed for image classification [10]. Here, we utilized CNN to classify facial images and the component images. Facial images contain holistic cues that can be extracted with the CNN. Similarly, Face component images such as left eyebrow, right eyebrow, left eye, right eye, nose, lips, mouth, and jaw also contains features that can be extracted with the CNN.

For all CNN models, we deployed 2D convolutional layers with 3 by 3 filters, with vary number of filters. All the convolutional layers contain ReLU activation function, with a max-pooling layer of 2 by 2 following the convolution operation. Due to the different image sizes, different models of the CNN must be deployed for each component. For instance, the facial image is standardized to 100 by 100 pixels can be convoluted 4 times with max-pooling following each operation. Meanwhile, for smaller images such as the left and right eyebrow, one can only apply the convolution with max-pooling operation twice.

After convolution, the outputs are flattened into an 1D array. ReLU is applied to the array to reduce the complexity. Next, we apply two fully connected layer with 4096 and 7 neurons, respectively. The final layer contains 7 neurons as there are 7 classes for classification. We included a drop-out of 0.5 between the fully connected layer to avoid overfitting. Finally, we applied a softmax activation function to yield the prediction output. The CNN parameters are optimized with Adam. Also, the training data is split via a 80:20% split into a small training and validation set for the CNN to evaluate and optimize the parameters. Training weights are implemented proportional to the number of training samples to account for the imbalanced dataset for training. This results in 9 CNN models, 9 predictions with 7 classes for each image.

In summary, face and jaw component uses a model that can convolute and max pool 4 times. Left eyebrow and right eyebrow component models are constructed with two sets of convolution and max pooling. Lips components use the least layers of two times convolution and one max pooling before flattening due to the dataset is small enough. Other facial components share another model with three sets of convolution and max pooling. Figures and summary models below show the different types of models used in this project for each component.

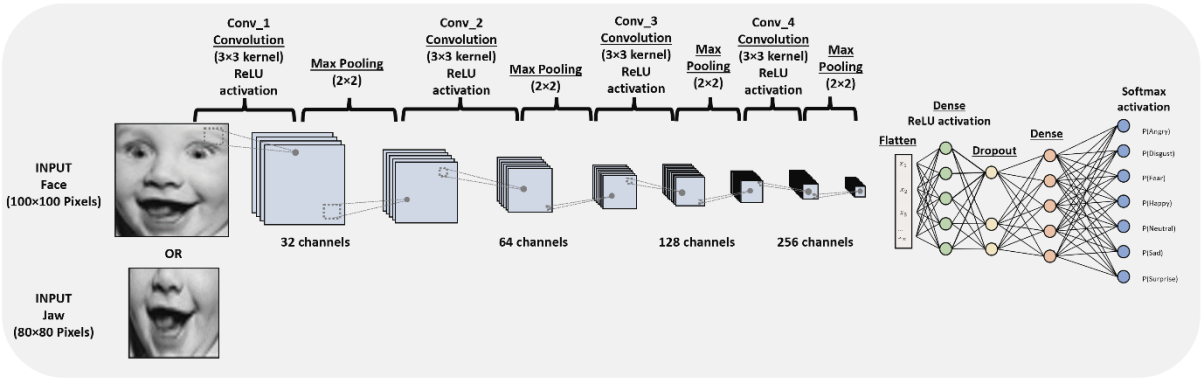


Figure 13: Proposed CNN model for face and jaw image.

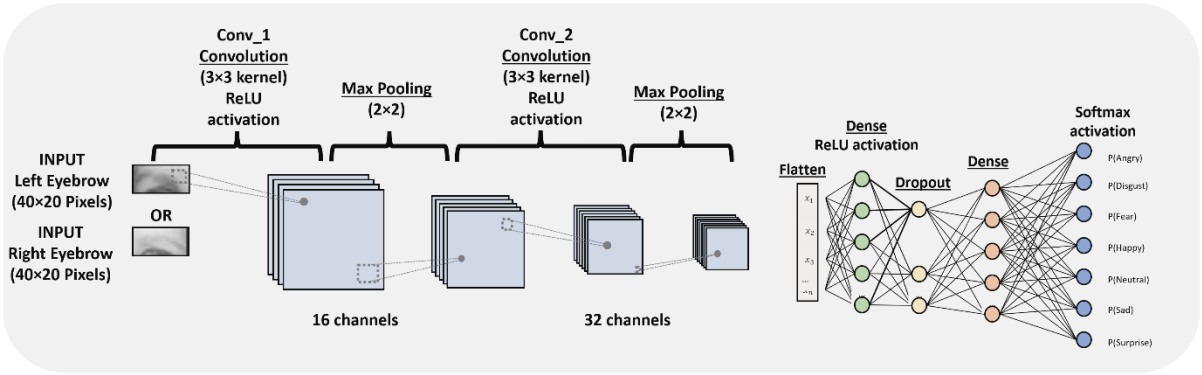


Figure 14: Proposed CNN model for eyebrow image.

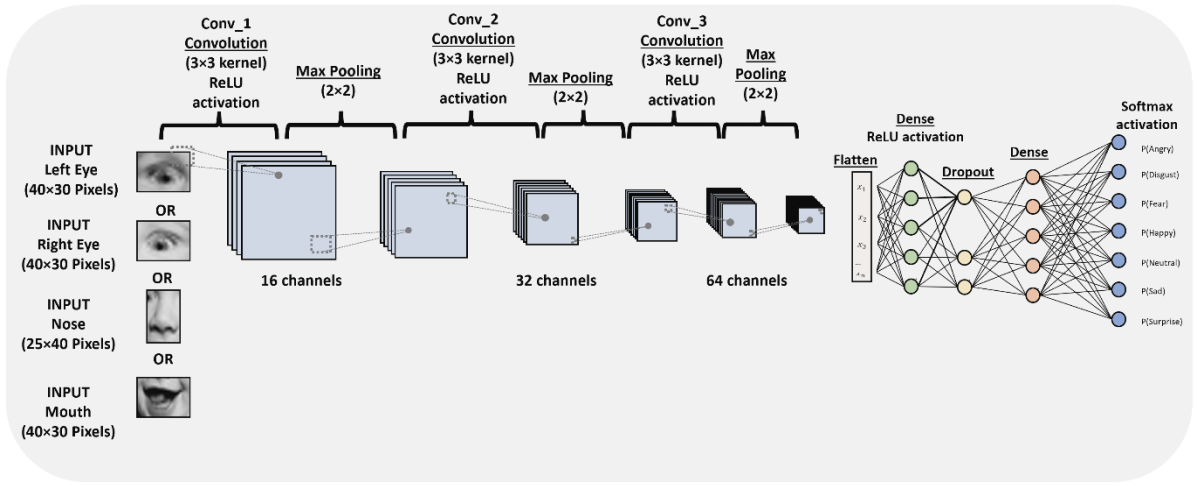


Figure 15: Proposed CNN model for eyes, nose, and mouth image.

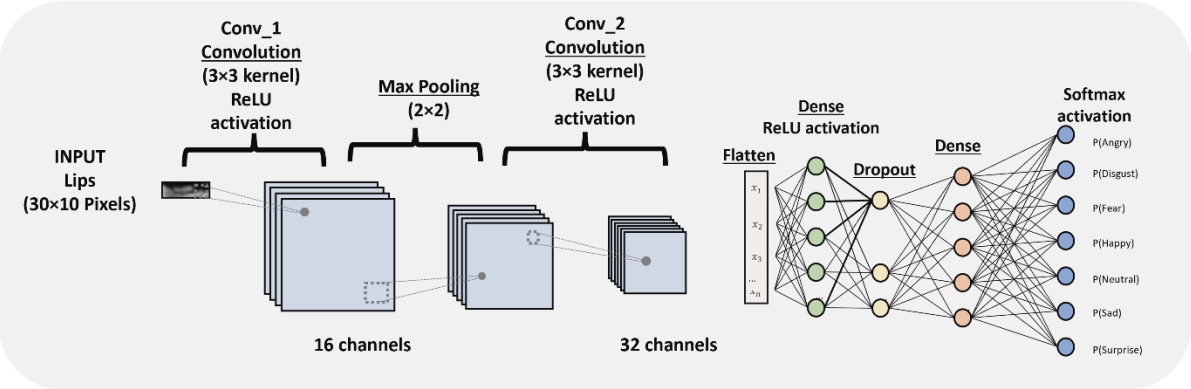


Figure 16: Proposed CNN model for lip image.

Catboost

CatBoost is an open-sourced machine learning algorithm based on gradient boosting techniques for classification and regression [4]. The term "Cat" arises from the word "category", as CatBoost was originally developed to process category data. In contrast to other machine learning model such as gradient boosting, XGBoost, LightGB, and others, CatBoost has the advantage of faster hyper-parameter tuning, which reduces the changes of overfitting. CatBoost has multiple parameters such as the number of trees, learning rate, regularization, tree depth, fold size, bagging temperature and others. In this study, we deploy CatBoost for classification of the landmark coordinate features.

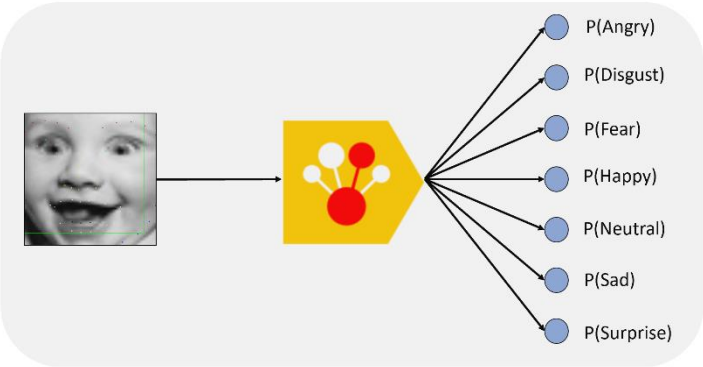


Figure 17: Catboost model for landmark features.

Chapter 4 Results

Confusion Matrix

After training the models with training dataset per discussed in Chapter 3, testing data is validated. To visually understand the data, we created a confusion matrix.

Holistic Cue Confusion Matrix

Figure 18 summaries the confusion matrix for holistic approach, which is taking the face to determine the face expression. The matrix is normalized to display the percentage of the testing sample. From here, we can see that the prediction is the best for happy class, followed by disgust and neutral. However, the result is not so good for fear and sad. Surprisingly, models misclassify the fear class more with the neutral class than the disgust class. Also, sad class is closely mis-predicted as neutral for the FER2013 dataset.

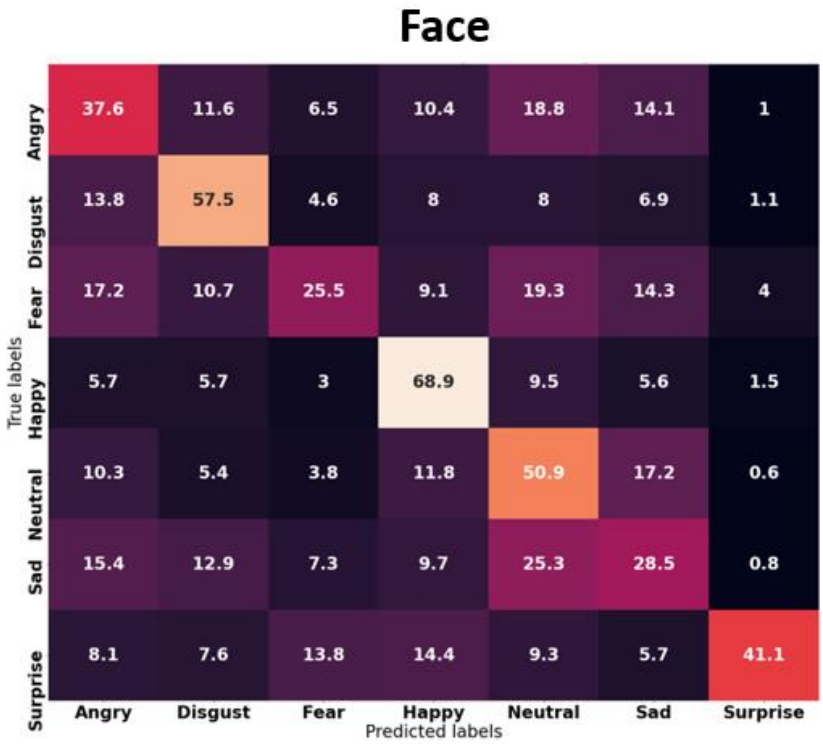


Figure 18: Confusion matrix for face image.

Componential Cue Confusion Matrix

For higher face components such as left and right eyebrows and eyes, the data is scattered all over the matrix. However, it does good in disgust and surprise expression. This is due to eyebrows being shrunk and lifted upwards and eyes being smaller while expressing these emotions.

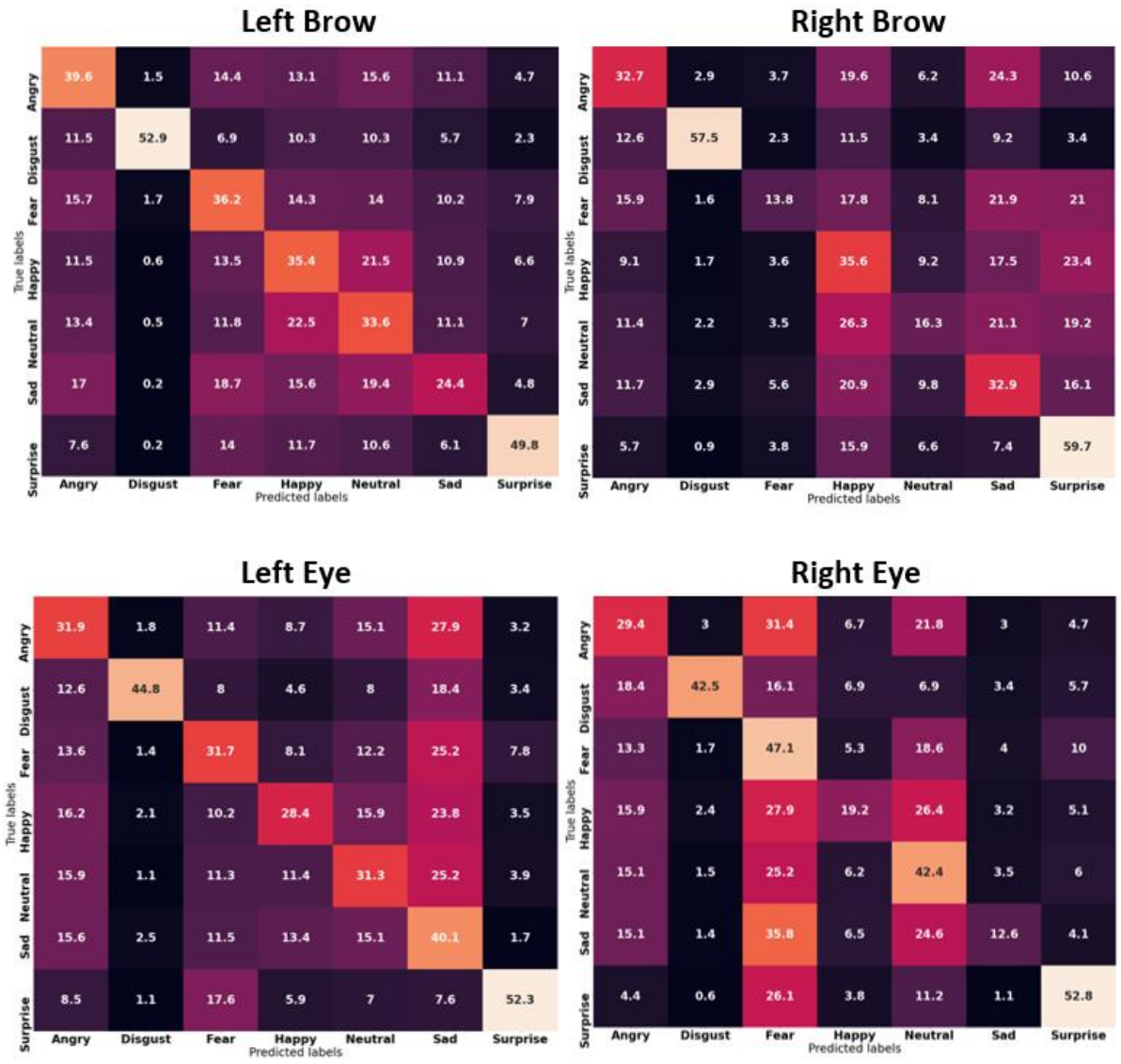


Figure 19: Confusion matrix for eyebrows and eyes images.

For middle and lower face components such as jaw, nose, lips and mouth, the testing dataset has confused result in nose data. This is understandable due to nose does not play a big factor in human's facial expression. For jaw, lips and mouth, the test data does the best prediction in happy and surprise. This may be due to mouth is open wide, teeth are shown and jaws stretching while expressing surprise expression. And for happy expression, the mouth ends being moved forward.

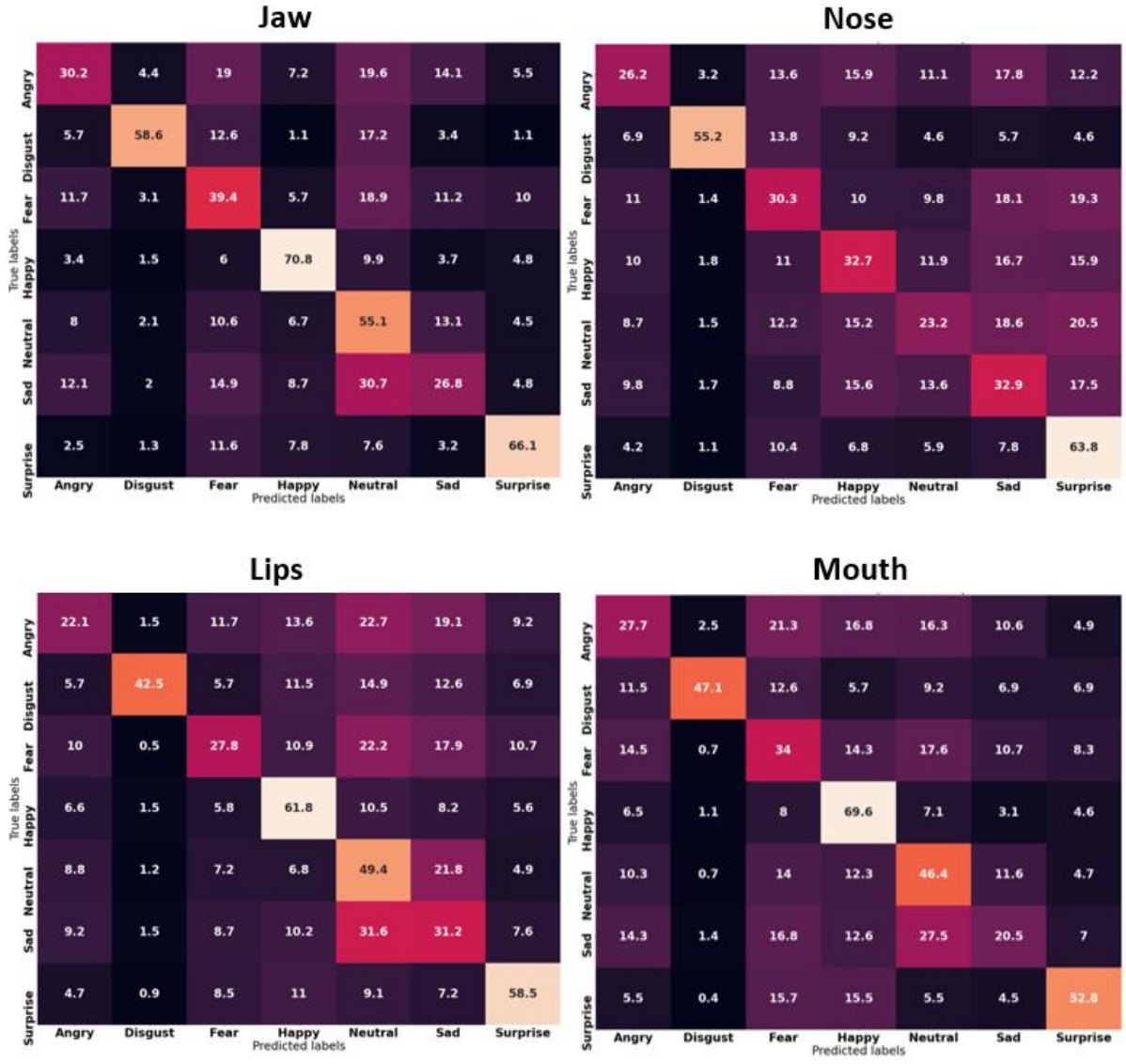


Figure 20: Confusion matrix for jaw, nose, lips, and mouth images.

Hybrid Cue Confusion Matrix

For 68 landmark coordinates, we can see that the data is the best in happy and surprise expression just like most of the other confusion matrices. Also for the neutral expression, the data is still strong enough to support more than fifty percent of the total testing data. For negative expressions such as angry, disgust, fear and sad, the data is not strong enough to be more than fifty percent but overall, we can see that it is predicted correctly most of the time.

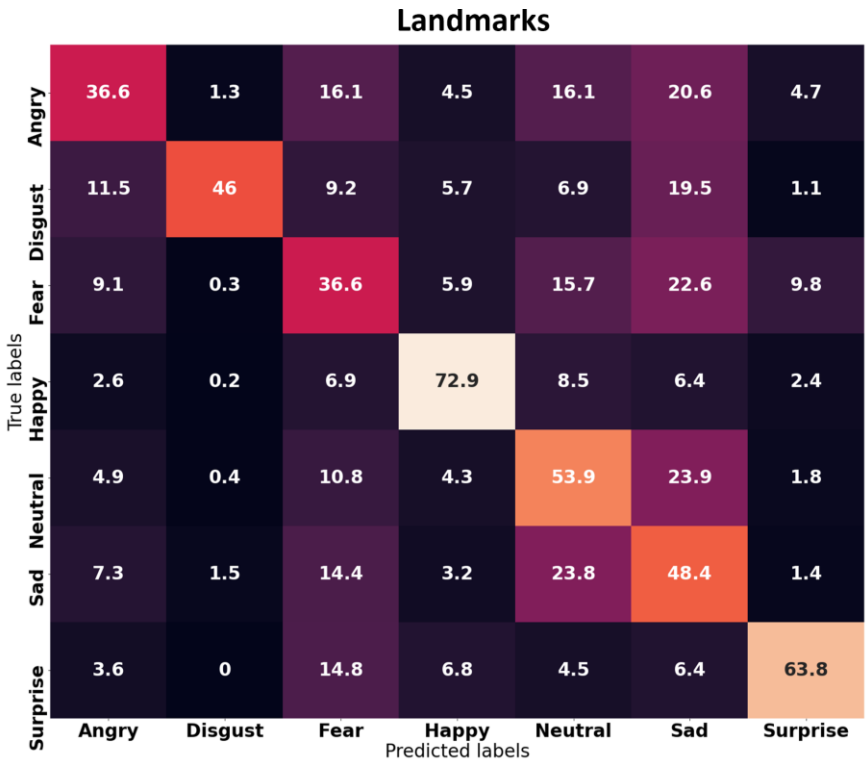


Figure 21: Confusion matrix for landmark features.

Combination of Componential and Holistic Cues

After collecting results from 9 models, both soft and hard voting is done. Hard voting has slightly better results compared to soft voting. Overall, happy, neutral and surprise class achieved the best results.

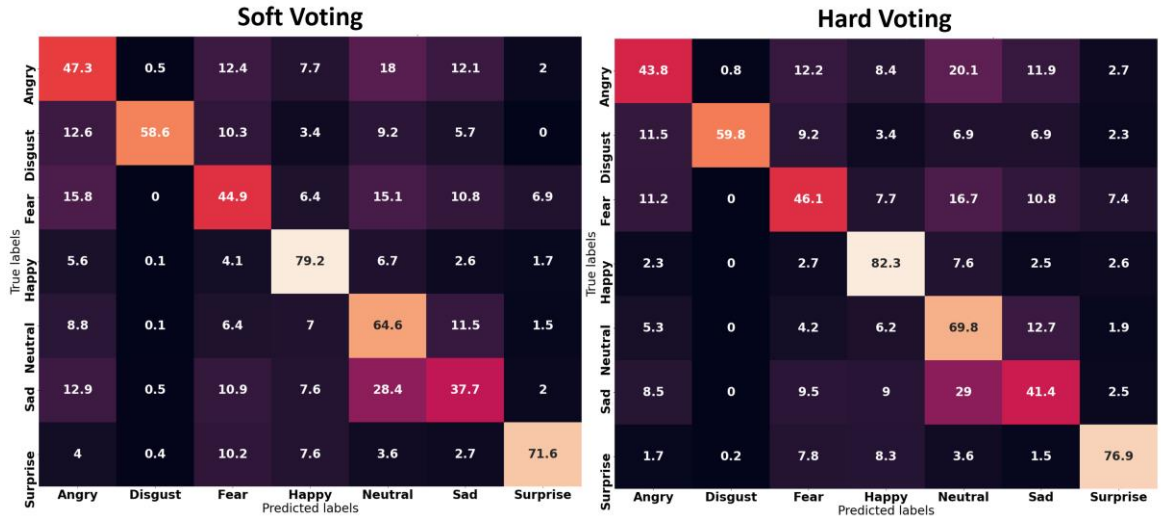


Figure 22: Confusion matrix for soft and hard voting model.

Performance Measure

To compare the performance across different models, we must deploy several performance metrics. The common metrics deployed are accuracy (ACC), precision (PRE), recall (REC), and F1-score (F1). However, as mentioned earlier, the dataset that is used in the project is an imbalanced dataset. Hence, accuracy is not the most optimal parameter. Instead, we focus on PRE, REC, and F1. The formulas to compute those metrics for a binary classification problem is listed below.

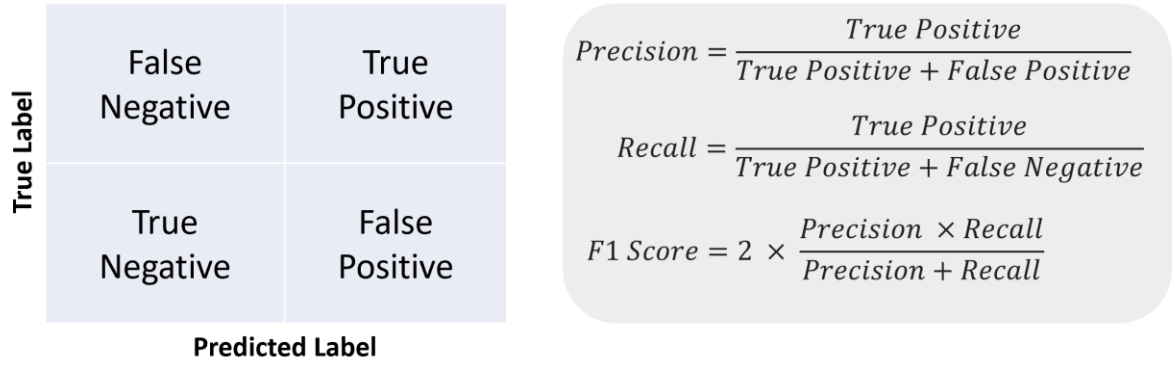


Figure 23: Performance Measure Labels.

Table 3: Accuracy of all the emotions obtained by the models for each component.

Metrics	Face	Jaw	Left Brow	Right Brow	Left Eye	Right Eye	Nose	Mouth	Lips	Landmark	Soft Voting	Hard Voting
Accuracy	0.472	0.520	0.363	0.317	0.344	0.319	0.338	0.465	0.452	**0.552**	0.639	0.612

Table 4: Precision of all the emotions obtained by the models for each component.

Emotion	Face	Jaw	Left Brow	Right Brow	Left Eye	Right Eye	Nose	Mouth	Lips	Landmark	Soft Voting	Hard Voting
Angry	0.354	0.401	0.318	0.318	0.248	0.244	0.305	0.303	0.303	**0.520**	0.518	0.457
Disgust	0.120	0.338	0.582	0.360	0.339	0.308	0.375	0.456	0.402	**0.625**	0.555	0.447
Fear	**0.392**	0.342	0.275	0.342	0.284	0.195	0.282	0.266	0.343	0.320	0.813	0.818
Happy	0.717	0.804	0.473	0.413	0.545	0.572	0.505	0.670	0.711	**0.858**	0.458	0.437
Neutral	0.439	0.442	0.311	0.316	0.347	0.313	0.334	0.446	0.393	**0.495**	0.537	0.536
Sad	0.284	0.324	0.265	0.210	0.209	**0.378**	0.234	0.291	0.247	0.325	0.897	0.836
Surprise	**0.783**	0.612	0.513	0.295	0.639	0.545	0.333	0.555	0.522	0.705	0.763	0.792
Average	0.354	0.401	0.318	0.318	0.248	0.244	0.305	0.303	0.303	**0.520**	0.518	0.457

Table 5: Recall of all the emotions obtained by the models for each component.

Emotion	Face	Jaw	Left Brow	Right Brow	Left Eye	Right Eye	Nose	Mouth	Lips	Landmark	Soft Voting	Hard Voting
Angry	0.376	0.302	**0.396**	0.327	0.319	0.294	0.262	0.277	0.221	0.366	0.438	0.473
Disgust	0.575	**0.586**	0.529	0.575	0.448	0.425	0.552	0.471	0.425	0.460	0.598	0.586
Fear	0.255	0.394	0.362	0.138	0.317	**0.471**	0.303	0.340	0.278	0.366	0.461	0.449
Happy	0.689	0.708	0.354	0.356	0.284	0.192	0.327	0.696	0.618	**0.729**	0.823	0.792
Neutral	0.509	0.551	0.336	0.163	0.313	0.424	0.232	0.464	0.494	**0.539**	0.698	0.646
Sad	0.285	0.268	0.244	0.329	0.401	0.126	0.329	0.205	0.312	**0.484**	0.414	0.377
Surprise	0.411	**0.661**	0.498	0.597	0.523	0.528	0.638	0.528	0.585	0.638	0.769	0.716
Average	0.443	0.496	0.389	0.355	0.372	0.351	0.378	0.426	0.419	**0.512**	0.600	0.577

Table 6: F1 of all the emotions obtained by the models for each component.

Emotion	Face	Jaw	Left Brow	Right Brow	Left Eye	Right Eye	Nose	Mouth	Lips	Landmark	Soft Voting	Hard Voting
Angry	0.365	0.344	0.353	0.323	0.279	0.267	0.282	0.289	0.256	**0.430**	0.490	0.460
Disgust	0.198	0.429	0.554	0.442	0.386	0.357	0.447	0.463	0.413	**0.530**	0.717	0.689
Fear	0.309	**0.366**	0.313	0.197	0.299	0.275	0.292	0.298	0.307	0.341	0.488	0.453
Happy	0.702	0.753	0.405	0.382	0.373	0.287	0.397	0.683	0.661	**0.788**	0.818	0.804
Neutral	0.471	0.491	0.323	0.215	0.329	0.360	0.274	0.455	0.438	**0.516**	0.607	0.586
Sad	0.285	0.293	0.254	0.257	0.274	0.189	0.274	0.241	0.276	**0.389**	0.435	0.405
Surprise	0.539	0.636	0.505	0.395	0.575	0.537	0.438	0.541	0.552	**0.670**	0.766	0.752
Average	0.410	0.473	0.387	0.316	0.360	0.325	0.343	0.424	0.415	**0.523**	0.617	0.593

We listed the ACC, PRE, REC, and F1 of each emotions for each models in Table 3, Table 4, Table 5, and Table 6, respectively.

Overall, the soft voting approach yield the highest ACC, achieving an ACC of 63.9%, with the hard voting approach yielding the second-best performance with an ACC of 61.2%. This is expected as the voting approach utilized outputs from all the individual component models.

Meanwhile, for an individual component model, the model that utilized the landmark coordinates achieved the highest ACC, outperforming all the CNN models. Among the CNN models, the CNN trained with jaw image yield the best ACC, but only obtained an ACC of 52.0%.

Next, we compare the PRE obtained from all the models. Overall, the landmark model attained the highest average PRE, outperforming even the soft and hard voting models. However, the soft and hard voting models achieved higher REC than the landmark model by a more significant amount. Consequently, the soft voting model achieved the highest average F1 score of 0.617, with the hard voting model behind at 0.593. Meanwhile, for the individual component models, the landmark yielded the highest score again, with a F1 of 0.523.

Overall, it is apparent that the landmark model outperformed all the CNN models, despite it being a traditional machine learning model. However, the landmark features are extracted with the 68-dlib landmark detector, which by itself is a supervised deep learning model. Hence, the features extracted with the 68-dlib landmark detector is highly accurate and valuable, which may outperform the CNN models we deployed here. However, from the results reported with the soft and hard voting models, the CNN models can also play an influential impact to the overall performance. Despite the initial abysmal results reported by most of the CNN models, their outputs are nonetheless valuable as it can improve overall results when combined via a voting system.

Chapter 5 Discussion

The best ACC obtained from our proposed model is 63.8%. as compared to existing studies [11] (see Table 7). We achieved the poorest ACC but is comparable to the Bag of Visual Words (BOVW) model. This is expected for several reasons. Firstly, one cannot implement the landmark detector on all the images in the FER2013 dataset. The 68-landmark detector requires all the landmarks to be present in an image to extract all the 68 landmarks, otherwise, it will produce an error. Unfortunately, most of the images from the FER2013 dataset are cropped in a way that parts of a face may be unavailable; the jaw may be out of frame, or the individual is looking to their side, making one eye out of view. Consequently, a significant amount of images in the FER2013 dataset must be discarded to deploy our pipeline as seen in Table 2. With much fewer data for training, we may have insufficient data to train a robust supervised model. As neural networks require a huge amount of training dataset to be adequate, our CNN models are unable to achieve the decent performance.

Moreover, the FER2013 dataset is extremely imbalanced, with significantly more happy class and less disgust class than the other emotions. Hence, reporting ACC as the sole performance metrics may not be appropriate. Nonetheless, most studies only reported the ACC.

Table 7: Accuracy reported by existing studies on the FER2013 dataset.

Model	ACC
Bag of Visual Words (BOVW)	65.07%
pre-trained VGG-face	65.65%
fine-tuned VGG-face	71.50%
fine-tuned VGG-f	69.38%
VGG-13	66.31%
CNNs and BOVW + global SVM	73.34%
CNNs and BOVW + local SVM	74.92%
Proposed Model	63.87%

Chapter 6 Implementation

To implement the pipeline on a working system, we utilized the webcam from a desktop or laptop. The overall graphical user interface (GUI) of the real-time emotion detector is created using OpenCV.

First, we preload the facial detector from dlib library, 68-dlib landmark detector, 9 CNN models trained with face and other facial components, and the CatBoost classifier. Then, we deployed OpenCV to enable the webcam to capture the live video from the camera of the system. For each frame of the video, we deployed the pretrained facial detector to identify any faces. If at least one face is detected, we perform the following:

1) Crop the detected face into an image.
2) Resize the image to a 100 by 100 pixels image.
3) Convert the image into black and white.
4) Extract the 68 landmarks using the 68-dlib landmark detector.
5) Extract the 8 facial components with the 68 landmarks.
6) Deploy the 9 pretrained CNN models to perform emotion prediction for the face and the 8 facial components to obtain the probability outputs.
7) Deploy the pretrained CatBoost classifier to perform emotion prediction for the landmark coordinates features to obtain the probability outputs.
8) Take a soft voting of all the 10 probability outputs to obtain the final probability outputs.
9) Take the argmax of the probability output to obtain the final emotion prediction.
10) Display the probability distribution for each emotion class using OpenCV.
11) Display a frame around the face detected and present the emotion and the probability above the frame.
12) If more than one face is detected, we repeat step 1 to 10 for that face, and change the colour code for the probability distribution, face frame, and annotation accordingly. The colour code for face 1, 2, 3, and 4 is red, green, blue, and yellow, respectively.
13) To close the webcam, press “q” to terminate.

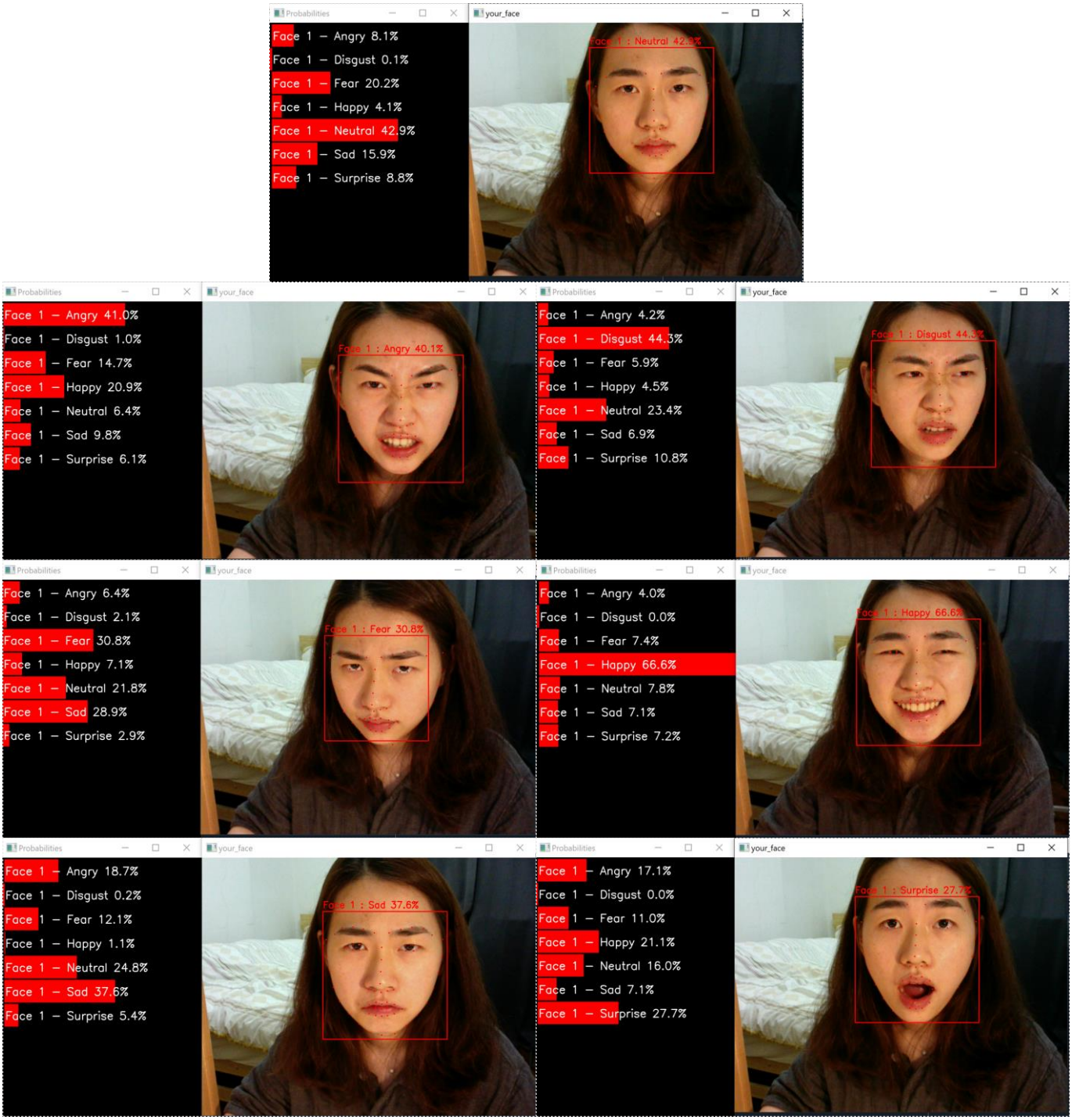


Figure 24: Webcam images for implementation of real-time face expression recognition.

Chapter 7 Conclusion

In this study, we performed emotion detection with facial components cues. We perform facial detection to obtain a facial image, before deploying the 68-landmark detector to extract 8 componential images and their respective landmark coordinates. Next, we performed CNN multi-class classification on the facial image and the 8 facial component images to predict 7 emotion classes: angry, disgust, fear, happy, neutral, sad, and surprise. Similarly, we deployed a CatBoost classifier to classify the landmark coordinates directly. Finally, we compiled the outputs of the 9 CNN models and the CatBoost model and applied soft and hard voting, to obtain the final results. The soft voting approach yield the best results, with an ACC of 63.8%, outperforming all the individual models alone.

For future work, we want to develop deeper CNNs and employ pretrained networks such as VGGNet, ResNet, or Inception to train the facial and facial componential images, to achieve better results. Concurrently, the results obtained from the CNNs are not as promising as those reported in literature. Additionally, we may utilize other datasets to obtain more training data to train a better emotion classification system. That way, we will have more training data that contains all the 68 landmarks, to enable us to train a more robust system to classify emotion more accurately.

Bibliography

[1] M. Sambare, “Kaggle,” [Online]. Available: https://www.kaggle.com/datasets/msambare/fer2013MANAS SAMBAR. [Accessed January 2021].

[2] A. Rosebrock, “Py Image Search,” April 2018. [Online]. Available: https://pyimagesearch.com/2018/04/02/faster-facial-landmark-detector-with-dlib/. [Accessed February 2021].

[3] I. José, “Towards Data Science,” June 2018. [Online]. Available: https://towardsdatascience.com/facial-mapping-landmarks-with-dlib-python-160abcf7d672. [Accessed February 2021].

[4] M. Przybyla, “Towards Data Science,” March 2021. [Online]. Available: https://towardsdatascience.com/4-easy-steps-for-implementing-catboost-c196fd82274b. [Accessed June 2021].

[5] M. Basavarajaiah, “Medium,” March 2019. [Online]. Available: https://medium.com/@bdhuma/6-basic-things-to-know-about-convolution-daef5e1bc411#:~:text=In%20image%20processing%2C%20convolution%20is,effect%20of%20the%20convolution%20process.. [Accessed February 2021].

[6] J. Brownlee, “Machine Learning Mastery,” April 2019. [Online]. Available: https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/#:~:text=Maximum%20pooling%2C%20or%20max%20pooling,the%20case%20of%20average%20pooling.. [Accessed May 2021].

[7] C. Maklin, “Towards Data Science,” June 2019. [Online]. Available: https://towardsdatascience.com/machine-learning-part-20-dropout-keras-layers-explained-8c9f6dc4c9ab. [Accessed May 2021].

[8] “Towards Data Science,” Nov 2018. [Online]. Available: https://towardsdatascience.com/softmax-function-simplified-714068bf8156. [Accessed May 2021].

[9] J. Brownlee, “Machine Learning Mastery,” April 2021. [Online]. Available: https://machinelearningmastery.com/voting-ensembles-with-python/. [Accessed May 2021].

[10] “Towards Data Science,” December 2018. [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53.

[11] M.-I. Georgescu, R. T. Ionescu and M. Popescu, “Local Learning with Deep and Handcrafted Features,” p. 10, 2020.